eDAM-8015
6-channel RTD Input Module
User's manual

Web site: www.inlog.com.tw

Trademark:
The names used in this manual for indentification only maybe registered trademarks of their respective companies

rev 1.2 July 11, 2011

Table of Contents

Chapter 1 Introduction 4

1.1 Overview 4
1.2 Communication and Programming 4
1.3 Software Configuration and Calibration 4
1.4 Watchdog Timer 4
1.5 Power Requirements 4

Chapter 2 About the eDAM Modules 5

2.1 Outline of eDAM modules 5
2.2 Module Dimension 5
2.3 Specifications 7
2.4 Block diagram of modules 8
2.5 EDAM-8015 wire connection 8
2.6 EDAM8015 pin assignments 9

Chapter 3 Installation 10

3.1 Set up an eDAM network 10
3.2 Host computer 10
3.3 Power supply 10
3.4 Communication Wiring 10
3.5 eDAM Utility Software 11
3.6 eDAM Isolated RS-232/RS485 Converter 11
3.7 Initialization Procedure 11
3.8 Install a New eDAM to a Existing Network 12

Chapter 4 ASCII Command Set 13

4.1 Introduction 13
4.2 Format of eDAM Commands 13
4.3 Calculate Checksum: 13
4.4 Response of Commands 14
4.5 Summary of Command Set 14
4.6 General Command Sets 14
4.7 Host Watchdog Command Sets 15
4.8 Configuration Tables 15
4.8.1 Default Settings 15
4.8.2 Baud rate setting (CC) 15
4.8.3 RTD Input Type Setting (TT) 16
4.8.4 Data Format Setting (FF) 16
4.8.5 Analog Input Type and Data Format Table 17
4.8.6 RTD Over Range/Under Range Reading with ASCII command protocol 18
4.8.7 RTD Over Range/Under Range Reading with Modbus RTU protocol 18
4.8.8 Protocol Switching 18
4.8.9 INIT Mode 19
4.9 Command description 20
4.9.1 %AANNTTCCFF Set Module Configuration 20
4.9.2 #** Synchronized sampling 20
4.9.3 #AA Read analog data 21
4.9.4 #AAN Read analog input from channel N 21
4.9.5 $AA0 Span calibration 21
4.9.6 $AA1 Offset calibration 22
4.9.7 $AA2 Read configuration status 22
4.9.8 $AA4 Read Synchronized data 23
4.9.9 $AA5 Read the module reset status 23
4.9.10 $AA5VV Enable/disable channels for multiplexing 24
4.9.11 $AA6 Read channel status 24
4.9.12 $AA7CiRrr Set channel type individually 25
4.9.13 $AA8Ci Read individual channel type 25
4.9.14 $AAB Read Channel Burnout Status 25
4.9.15 $AAF Read the firmware version 26
4.9.16 $AA1 Read the INIT terminal status 26
4.9.17 $AAM Read the module name 26

4.9.18 $AAP Read the communication protocol ····27
4.9.19 $AAPN Set the communication protocol ····27
4.9.20 $AAS1 Reload default calibration parameters ····27
4.9.21 ~AAD Read the miscellaneous settings ····28
4.9.22 ~AADVV Set the miscellaneous settings ····28
4.9.23 ~AAEV Enable/disable calibration ····29
4.9.24 ~AAI Soft INIT ····29
4.9.25 ~AAO(Name) Set the module name ····29
4.9.26 ~AATnn Set the soft INIT timeout ····30
4.9.27 ~** Sends a "Host OK" ····30
4.9.28 ~AA0 Read the host watchdog status ····30
4.9.29 ~AA1 Reset the host watchdog status ····30
4.9.30 ~AA2 Read the host watchdog time out value ····31
4.9.31 ~AA3ETT Enables/disables the host watchdog ····31

Chapter 5 Modbus RTU Command structure ····32

5.1 Modbus Function code introductions ····32
5.2 Modbus Function code 0x46 for Module configuration ····32
5.2.1 Sub-function 00 (0x00) Read module name ····33
5.2.2 Sub-function 04 (0x04) Set module address ····34
5.2.3 Sub-function 05 (0x05) Read communication settings ····35
5.2.4 Sub-function 06 (0x06) Set communication settings ····36
5.2.5 Sub-function 07(0x07) Read channel type code ····37
5.2.6 Sub-function 08(0x08) Set channel type code ····38
5.2.7 Sub-function 32(0x20) Read firmware version ····39
5.2.8 Sub-function 37(0x25) Read channel enable/disable status ····40
5.2.9 Sub-function 38(0x26) Set channel enable/disable ····41
5.2.10 Sub-function 41(0x29) Read miscellaneous settings ····42
5.2.11 Sub-function 42(0x2A) Set miscellaneous settings ····43
5.3 Modbus Address Mapping ····44
5.3.1 Discrete address (1-bits) ····44
5.3.2 Register address (16-bits) ····45
5.4 Modbus Engineering Data Format Table ····46
5.5 Modbus Hex 2's complement Data Format Table ····47

Chapter 1 Introduction

1.1 Overview

The eDAM-*8000* analog modules is a set of intelligent sensor to computer interface modules containing built-in microprocessor. They provide data comparison, and digital communication functions. Some modules provide analog I/O lines for controlling and monitoring analog signals.

The common features of the EDAM-8015 modules are as follows:

1. 3000V DC inter-module isolation
2. Support both ASCII and Modbus/RTU protocol
3. 24-bit sigma-delta ADC to provide excellent accuracy
4. Direct RTD connection
5. Software calibration
6. Supported RTD types are as follows:
   - Platinum, 100 Ohms at 0°C, α= 0.00385
   - Platinum, 100 Ohms at 0°C, α= 0.003916
   - Platinum, 1000 Ohms at 0°C, α= 0.00385
   - Nickel, 120 Ohms at 0°C, α= 0.00672
   - Copper, 100 Ohms at 0°C, α= 0.00421
   - Copper, 1000 Ohms at 0°C, α= 0.00421
   - Copper, 100 Ohms at 25°C, α= 0.00427
   - Copper, 50 Ohms at 0°C
   - Nickel, 100 Ohms at 0°C

1.2 Communication and Programming

eDAM modules can connect to and communicate with all computers and terminals. They use RS-485 transmission standards, and communicate with ASCII format commands. All communications to and from the module are performed in ASCII, which means that eDAM modules can be programmed in virtually any high-level language.

Up to 256 eDAM modules may be connected to an RS-485 multi-drop network by using the eDAM RS-485 repeater, extending the maximum communication distance to 4,000 ft.

1.3 Software Configuration and Calibration

EDAM modules contain no pots or switches to set. By merely issuing a command from the host computer, you can change an analog input module to accept several ranges of voltage input.

Remote configuration can be done by using the command set's configuration and calibration commands.

By storing configuration and calibration parameters in a nonvolatile EEPROM, modules are able to retain these parameters in case of power failure.

1.4 Watchdog Timer

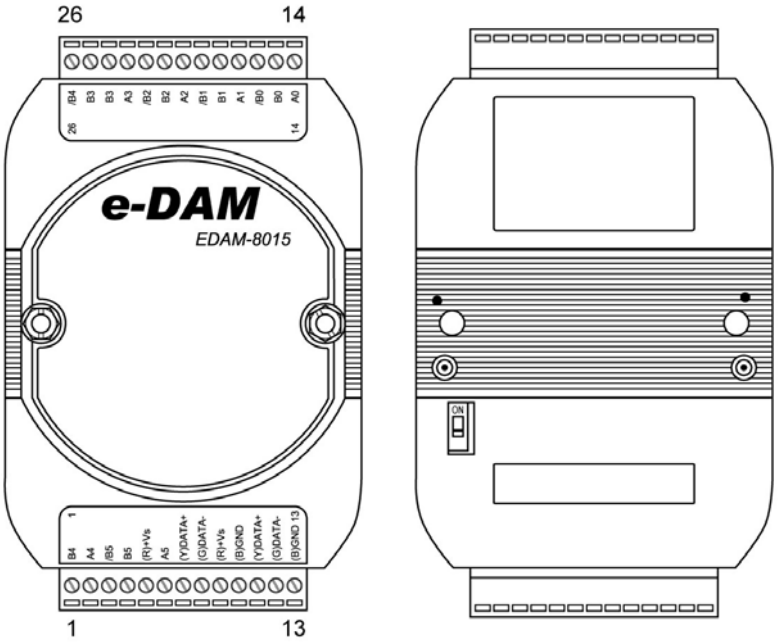
A watchdog timer supervisory function will automatically reset the eDAM modules in the event of system failure. Maintenance is thus simplified.

1.5 Power Requirements

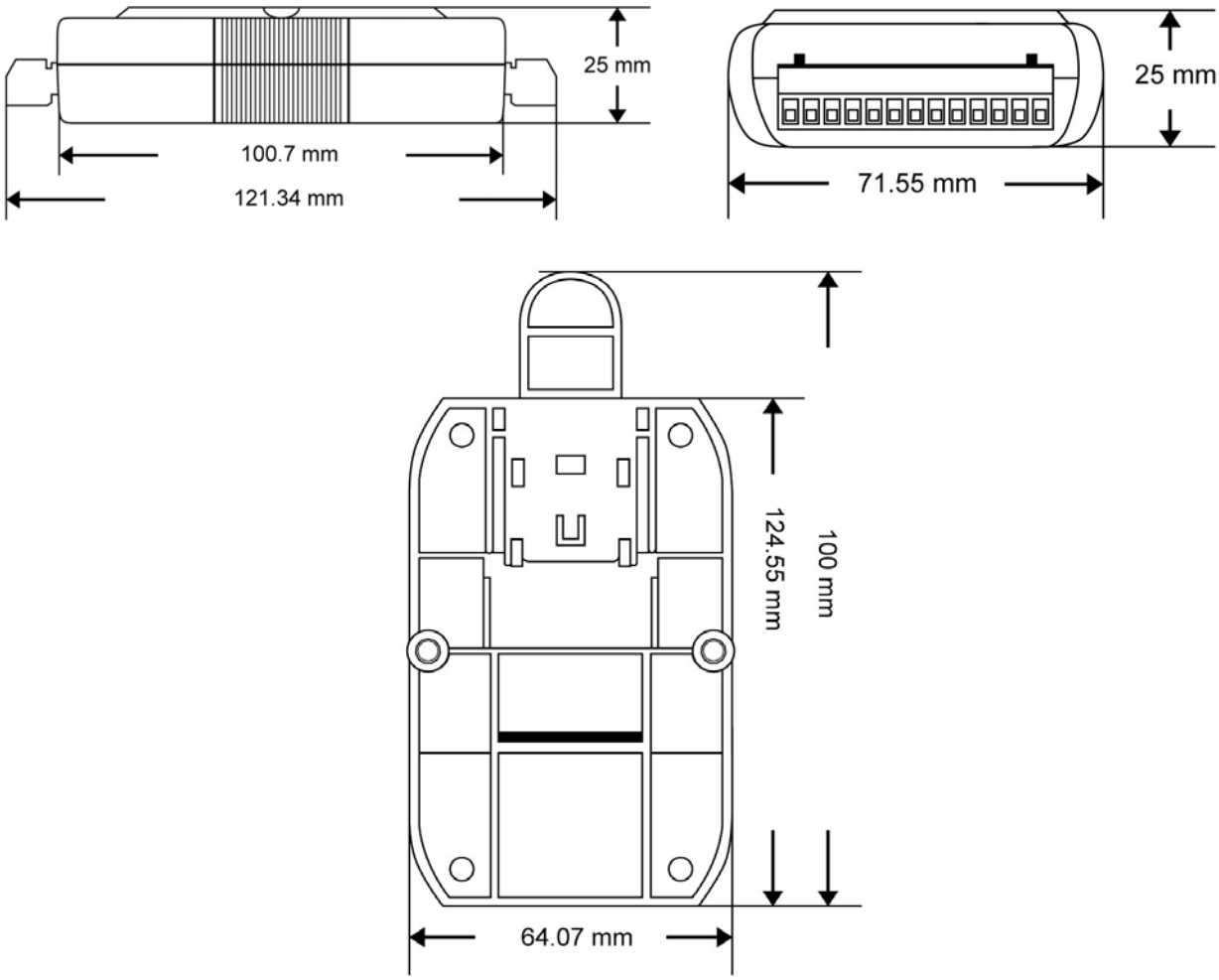
Although the modules are designed for standard industrial unregulated 24V DC power supply , they accept any power unit that supplies power within the range of +10 to +30 V DC . The power supply ripple must be limited to 5 V peak-to-peak, and the immediate ripple voltage should be maintained between +10 and +30 V DC .

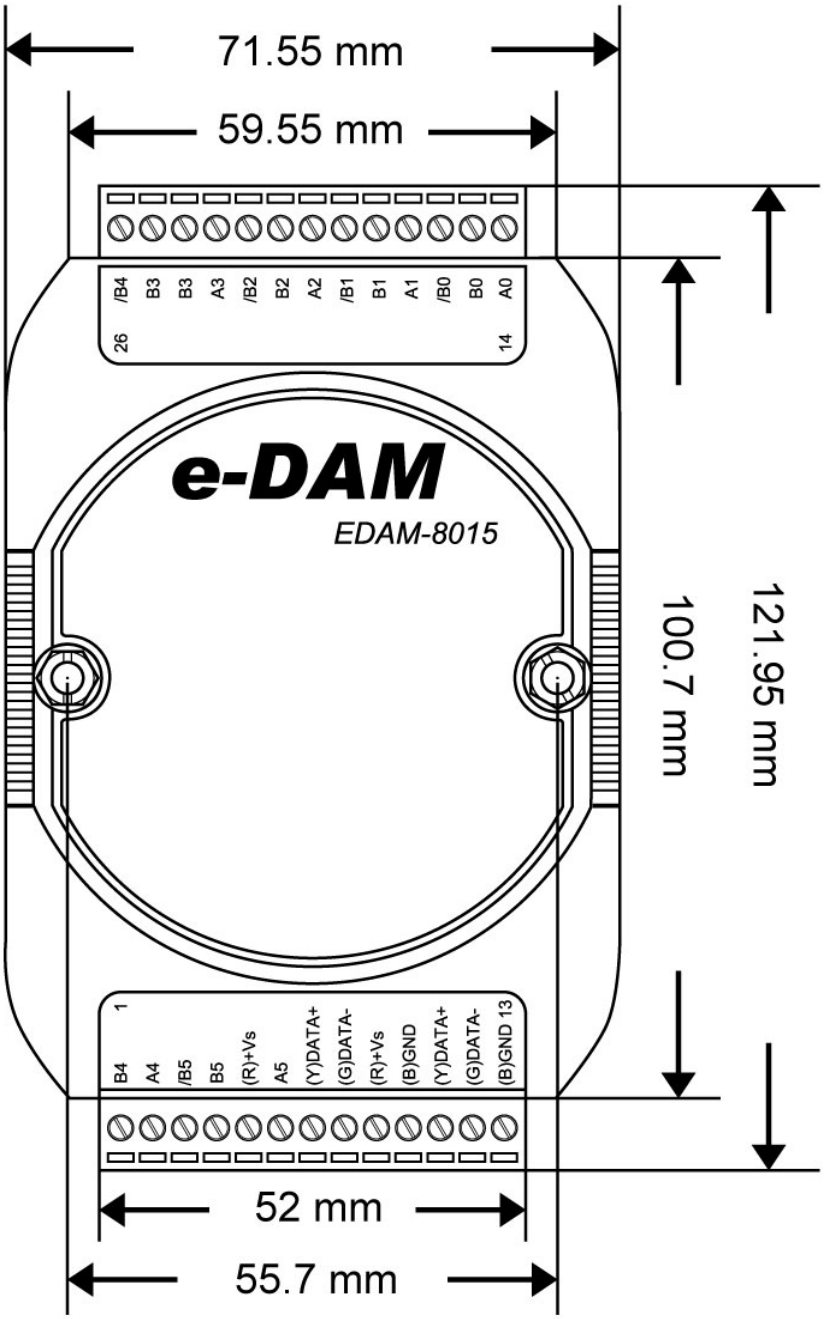
Chapter 2 About the eDAM Modules

2.1 Outline of eDAM modules



2.2 Module Dimension

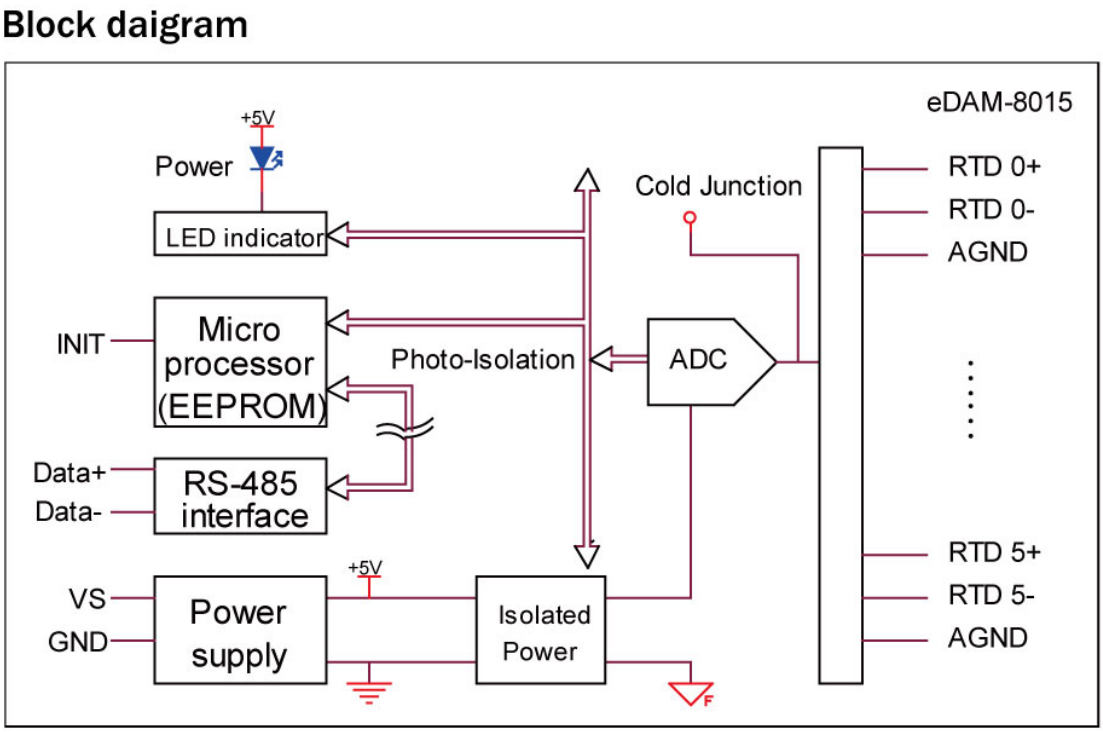



71.55 mm
59.55 mm
/B4 B3 B3 A3 /B2 B2 A2 /B1 B1 A1 /B0 B0 A0
26
14
e-DAM
EDAM-8015
100.7 mm
121.95 mm
1
B4 A4 /B5 B5 (R)+Vs A5 (Y)DATA+ (G)DATA- (R)+Vs (B)GND (Y)DATA+ (G)DATA- (B)GND 13
52 mm
55.7 mm

2.3 Specifications

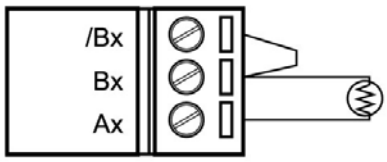
- Input channels: 6 RTD inputs.
- Interface: RS-485, 2 wires
- Speed (bps): 1200, 2400, 4800, 9600, 19.2K, 38.4K , 115.2K
- Analog Input type: Differential input
- Analog Channels Numbers: 6
- Analog Resolution: 16 bits
- Sampling Rate :10 Samples/Second
- Bandwidth : 15.7 Hz
- Accuracy : ±0.1%
- Zero Drift : 0.5μV/°C
- Span Drift : 25ppm/°C
- CMR@50/60Hz : 150dB
- NMR@50/60Hz : 100dB
- Input Impedance :10M Ohms
- RTD wires: 2 /3 /4 wires
- RTD types: Platinum, 100 Ohms at 0°C, α= 0.00385
  Platinum, 100 Ohms at 0°C, α= 0.003916
  Platinum, 1000 Ohms at 0°C, α= 0.00385
  Nickel, 120 Ohms at 0°C, α= 0.00672
  Copper, 100 Ohms at 0°C, α= 0.00421
  Copper, 1000 Ohms at 0°C, α= 0.00421
  Copper, 100 Ohms at 25°C, α= 0.00427
  Copper, 50 Ohms at 0°C
  Nickel, 100 Ohms at 0°C
- Power supply: +10V to +30V

2.4 Block diagram of modules

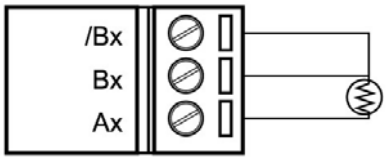


2.5 EDAM-8015 wire connection

4-wires RTD connection



3-wires RTD connection



2.6 EDAM8015 pin assignments

pin	name	description
1	B4	RTD Sense- of channel 4
2	A4	RTD Sense+ of channel 4
3	/B5	Execution current of channel 5
4	B5	RTD Sense- of channel 5
5	A5	RTD Sense+ of channel 5
6	DATA+	signal, positive
7	DATA-	signal, negative
8	+VS	+10V ~ +30Vdc
9	GND	Ground
10	DATA+	signal, positive
11	DATA-	signal, negative
12	+VS	+10V ~ +30Vdc
13	GND	Ground
14	A0	RTD Sense+ of channel 0
15	B0	RTD Sense- of channel 0
16	/B0	Execution current of channel 0
17	A1	RTD Sense+ of channel 1
18	B1	RTD Sense- of channel 1
19	/B1	Execution current of channel 1
20	A2	RTD Sense+ of channel 2
21	B2	RTD Sense- of channel 2
22	/B2	Execution current of channel 2
23	A3	RTD Sense+ of channel 3
24	B3	RTD Sense- of channel 3
25	/B3	Execution current of channel 3
26	/B4	Execution current of channel 4

Chapter 3 Installation

This chapter provides guidelines to what is needed to set up and install an eDAM network. A quick hookup scheme is provided that lets you configure modules before they are installed in a network.

To help you to connect eDAM modules with sensor inputs, several wiring examples are provided. Finally, you will find at the end of this chapter a programming example using the eDAM command set.

Be sure to carefully plan the layout and configuration of your network before you start. Guidelines regarding layout are given in Appendix E: RS-485 Network.

NOTICE: Except for changing eDAM to other compatible modules, which have on-board switches for their baud rate setting, eDAM modules should not be opened. There is no need to open the eDAM modules: all configuration is done remotely and there are no user serviceable parts are inside. Opening the cover will therefore void the warranty.

3.1 Set up an eDAM network

The following list gives an overview of what is needed to setup, install and configure an eDAM environment.

A host computer that can output ASCII characters with an RS-232C or RS-485 port.

Power supply for the eDAM modules (+10 to +30 V_{DC})

eDAM Series Utility software

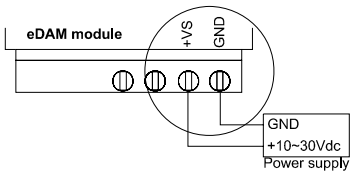
3.2 Host computer

Any computer or terminal that can output in ASCII format over either RS-232 or RS-485 can be connected as the host computer. When only RS-232 is available, an eDAM-8520 module (RS-232/RS-485 converter) is required to transform the host signals to the correct RS-485 protocol. The converter also provides opto-isolation and transformer-based isolation to protect your equipment.

For the ease of use in industrial environments the eDAM modules are designed to accept industry standard +24 V_{DC} unregulated power. Operation is guaranteed when using any power supply between +10 and +30 V_{DC}. Power ripples must be limited to 5 V peak to peak while the voltage in all cases must be maintained between +10 and +30 V_{DC} . All power supply specifications are referenced at module connector. When modules are powered remotely, the effects of line voltage drops must be considered.

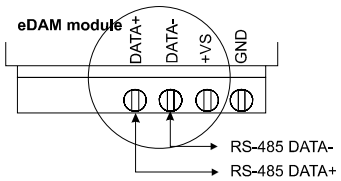
3.3 Power supply

All modules use on-board switching regulators to sustain good efficiency over the 10-30V input range, therefore we can assume that the actual current draw is inversely proportional to the line voltage. The following example shows how to calculate the required current that a power supply should be able to provide.



3.4 Communication Wiring

We recommend that shielded-twisted-pair cables that comply with the EIA RS-485 standard be used with the eDAM network to reduce interference.



3.5 eDAM Utility Software

A menu-driven utility program called "DOSEDAM.EXE" for DOS or "WINEDAM.EXE for Windows is provided for eDAM module configuration, monitoring and calibration. It also includes a terminal emulation program that lets you easily communicate through the eDAM command set

3.6 eDAM Isolated RS-232/RS485 Converter

When the host computer or terminal has only a RS-232 port, an eDAM-8520 Isolated RS-232/RS-485/422 converter connected to the host's RS-232 port is required.

This module equips a "Auto baud rate detector" inside, therefore it can detect the baud rate and data format automatically and control the direction of RS-485 precisely

3.7 Initialization Procedure

1. Power off the host computer and the installed eDAM-8520 to COM port of host computer.
2. Connect a brand new eDAM module with the RS-485. Set the module in *Default State* by shorting the **INIT*** pin to GND. Refer to Figure 1 for detailed wiring.
3. Power on the power supply for eDAM modules.
4. Use the eDAM utility to configure the address ID, baud rate, check-sum status and command sets of the module.

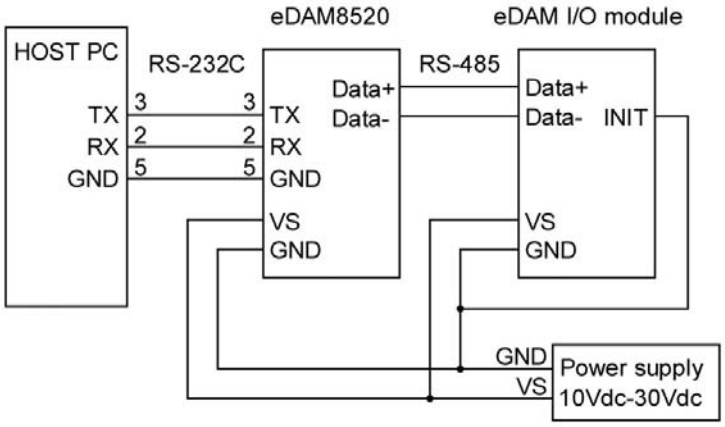


Figure 1

3.8 Install a New eDAM to a Existing Network

1. Equipments for Install a New Module
2. A existing eDAM network
3. New eDAM modules.
4. Power supply (+10 to +30 VDC)
5. Installing Procedures
6. Configure the new eDAM module according to the initialization procedure in section 3.7
7. The baud rate and check-sum status of the new module must be identity with the existing RS-485 network. The address ID must not be conflict with other eDAM modules on the network.
8. Power off the eDAM power supply of the existing RS-485 network.
9. Wire the power lines for the new eDAM with the existing network. Be careful about the signal polarity as wiring.
10. Wire the RS-485 data lines for the new eDAM with the existing network. Be careful about the signal polarity as wiring.
11. Wire to the input or output devices.
12. Power on the eDAM local power supply.
13. Use the eDAM utility to check entire network.

Chapter 4 ASCII Command Set

4.1 Introduction

The eDAM command is composed by numbers of characteristics, including the leading code, address ID, the variables, the optional check-sum byte, and a carriage return to indicate the end of a command.

The host computer can only command only one eDAM module except those synchronized commands with wildcard address command "#**". The eDAM may or may not give response to the command. The host should check the response to handshake with the modules.

4.2 Format of eDAM Commands

Syntax: (Leading code)(Addr)(Command)[Data] <Cksum><CR>

Every command begins with a delimiter character. There are five valid characters: a dollar sign $, a pound sign #, a percentage, a wave sign '~' ,sign % and an at sign @.

The delimiter character is followed by a two-character address (hexadecimal) that specifies the target module. The actual two character command follows the address. Depending on the command, an optional data segment follows the command string. An optional two character checksum may be appended to the total string. Every command is terminated by a carriage return (cr).

Conventions

Leading Code	The first characteristic of the eDAM command, such as %,$,#,~, @, ...etc(1- character)
Addr	Module's address ID, the value is in the range of 00 – FF (Hex) 2- character
Command	Command codes or value of variables
Data	Data needed by some output command
Checksum	Checksum in brackets indicate optional parameter, only checksum is enable then this field is required (2- character)
<CR>	carriage return(0x0D)
Note: all commands should be issued in ASCII uppercase characters. There is no spacing between characters.	

4.3 Calculate Checksum:

1. Calculate ASCII sum of all characters of command (or response) string except the character return(cr)
2. Mask the sum of string with 0ffh
3. [Checksum]={(Leading code)+(addr)+(command)+[data]} MOD 0x100

Example:

Command string: $012(cr)

Sum of string='$'+'0'+'1'+'2'=24h+30h+31h+32h=B7h

The checksum is B7h, and [CHK]="B7"

Command string with checksum=$012B7(cr)

Response string: !01400600(cr)

Sum of string='!'+'0'+'1'+'4'+'0'+'0'+'6'+'0'+'0'

=21h+30h+31h+34h+30h+30h+36h+30h+30h=1ACh

The checksum is ACh, and [CHK]="AC"

Response string with checksum=!01400600AC(cr)

4.4 Response of Commands

The response message depends on eDAM command. The response is also composed with several characteristics, including leading code, variables, and carriage return for ending. There are two kinds of leading code for response message, "!" or ">" means valid command and "?" means invalid. By checking the response message, user can monitor the command is valid or invalid.

But under the following conditions, there will have no response message.

- The specified address ID does not exist.
- Syntax error or communication error
- Some special commands do not have response.

4.5 Summary of Command Set

There are four categories of eDAM commands. The first is the *eDAM special command*s. The second is the *general command*s, the third is the *analog commands*, the forth is the *digital commands* and the last is *linear mapping commands*. All the commands used in the eDAM analog input module are list in the following table.

4.6 General Command Sets

Command	Response	Description	Sec.
%AANNTTCCFF	!AA	Sets the module configuration	4.9.1
#**	No Response	Synchronized sampling	4.9.2
#AA	>(Data)	Reads the analog inputs of all channels	4.9.3
#AAN	>(Data)	Reads the analog input of a channel	4.9.4
$AA0	!AA	Performs a span calibration	4.9.5
$AA1	!AA	Performs a zero calibration	4.9.6
$AA2	!AANNTTCCFF	Reads the module configuration	4.9.7
$AA4	>AAS(Data)	Reads the synchronized data	4.9.8
$AA5	!AAS	Reads the module reset status	4.9.9
$AA5VV	!AA	Enables/Disables a channel	4.9.10
$AA6	!AAVV	Reads the channel enabled/disabled status	4.9.11
$AA7CiRrr	!AA	Sets a single channel range configuration	4.9.12
$AA8Ci	!AACiRrr	Reads a single channel range configuration	4.9.13
$AAB	!AANN	Reads the channel diagnostic status	4.9.14
$AAF	!AA(Data)	Reads the firmware version	4.9.15
$AAI	!AAS	Reads the INIT terminal status	4.9.16
$AAM	!AA(Data)	Reads the module name	4.9.17
$AAP	!AASC	Reads the communication protocol	4.9.18
$AAPN	!AA	Sets the communication protocol	4.9.19
$AAS1	!AA	Reload default calibration parameters	4.9.20
~AAD	!AAVV	Reads the miscellaneous settings	4.9.21
~AADVV	!AA	Sets the miscellaneous settings	4.9.22
~AAEV	!AA	Enables/Disables calibration	4.9.23
~AAI	!AA	Soft INIT	4.9.24
~AAO(Name)	!AA	Sets the module name	4.9.25
~AATnn	!AA	Sets the soft INIT timeout	4.9.26

4.7 Host Watchdog Command Sets

Command	Response	Description	Sec.
~**	No Response	Host is OK	4.9.27
~AA0	!AASS	Reads the host watchdog status	4.9.28
~AA1	!AA	Resets the host watchdog status	4.9.29
~AA2	!AAETT	Reads the host watchdog timeout settings	4.9.30
~AA3ETT	!AA	Sets the host watchdog timeout settings	4.9.31

4.8 Configuration Tables

4.8.1 Default Settings

Default settings for the M-7033/33D and M-7015/15P modules are as follows:

- Protocol: ASCII command protocol
- Module Address: 01
- RTD Type: Type 20, Pt100, -100°C to 100°C
- Baud Rate: 9600 bps
- Filter set at 60Hz rejection

4.8.2 Baud rate setting (CC)

Code (CC)	03	04	05	06	07	08	09	0A
Baud rate	1200	2400	4800	9600	19200	38400	57600	115200

Note: The data bits are fixed at one start bit, eight data bits, no parity and one stop bit

4.8.3 RTD Input Type Setting (TT)

Type Code	Temperature Sensor Type	Temperature Range °C
20	Platinum 100, α= 0.00385	-100 ~ 100
21	Platinum 100, α= 0.00385	0 ~ 100
22	Platinum 100, α= 0.00385	0 ~ 200
23	Platinum 100, α= 0.00385	0 ~ 600
24	Platinum 100, α= 0.003916	-100 ~ 100
25	Platinum 100, α= 0.003916	0 ~ 100
26	Platinum 100, α= 0.003916	0 ~ 200
27	Platinum 100, α= 0.003916	0 ~ 600
28	Nickel 120	-80 ~ 100
29	Nickel 120	0 ~ 100
2A	Platinum 1000, α= 0.00385	-200 ~ 600
2B	Cu 100 @ 0°C,α= 0.00421	-20 ~ 150
2C	Cu 100 @ 25°C,α= 0.00427	0 ~ 200
2D	Cu 1000 @ 0°C,α= 0.00421	-20 ~ 150
2E	Platinum 100, α= 0.00385	-200 ~ 200
2F	Platinum 100, α= 0.003916	-200 ~ 200
80	Platinum 100, α= 0.00385	-200 ~ 600
81	Platinum 100, α= 0.003916	-200 ~ 600
82	Cu 50 @ 0°C	-50 ~ 150
83	Nickel 100	-60 ~ 180

4.8.4 Data Format Setting (FF)

7	6	5	4	3	2	1	0
FS	**CS**	**reserved**				**DF**	

Key	Description
DF	Data format 00: Engineer unit 01: % of FSR (full scale range) 10: 2's complement hexadecimal
CS	Check sum 0: disabled 1: enabled
FS	Filter settings 0: 60 Hz rejection 1:50 Hz rejection

4.8.5 Analog Input Type and Data Format Table

Type Code	RTD Type	Data Format	+F.S.	-F.S.
20	Platinum 100 α = 0.00385 -100 ~ 100°C	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+138.50	+060.60
21	Platinum 100 α = 0.00385 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+100.00
		2's comp HEX	7FFF	0000
		Ohms	+138.50	+100.00
22	Platinum 100 α = 0.00385 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+175.84	+100.00
23	Platinum 100 α = 0.00385 0 ~ 600°C	Engineering unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+313.59	+100.00
24	Platinum 100 α = 0.003916 -100 ~ 100°C	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+139.16	+060.60
25	Platinum 100 α = 0.003916 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+139.16	+100.00
26	Platinum 100 α = 0.003916 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+177.14	+100.00
27	Platinum 100 α = 0.003916 0 ~ 600°C	Engineering unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+317.28	+100.00
28	Nickel 120 -80 ~ 100°C	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+200.64	+066.60
29	Nickel 120 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+200.64	+120.60
2A	Platinum 1000 α = 0.00385 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+3137.1	+0185.2
2B	Cu 100 α = 0.00421 -20 ~ 150°C	Engineering unit	+150.00	-020.00
		% of FSR	+100.00	-013.33
		2's comp HEX	7FFF	EEEF
		Ohms	+163.17	+091.56
2C	Cu 100 α = 0.00427 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+167.75	+090.34
2D	Cu 1000 α = 0.00421 -20 ~ 150°C	Engineering unit	+150.00	-020.00
		% of FSR	+100.00	-013.33
		2's comp HEX	7FFF	EEEF
		Ohms	+1631.7	+0915.6

2E	Platinum 100 α = 0.00385 -200 ~ 200°C	Engineering unit	+200.00	-200.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+175.84	+018.49
2F	Platinum 100 α = 0.003916 -200 ~ 200°C	Engineering unit	+200.00	-200.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+177.14	+017.14
80*3	Platinum 100 α = 0.00385 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+313.59	+018.49
81*3	Platinum 100 α = 0.003916 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+317.28	+017.14
82*4	Cu 50 -50 ~ 150°	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+082.13	+039.24
83*5	Nickel 100 -60 ~ 180°C	Engineering unit	+180.00	-060.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+223.10	+069.50

4.8.6 RTD Over Range/Under Range Reading with ASCII command protocol

Data Format	Over Range	Under Range
Engineering Unit	+9999.9	-9999.9
% of FSR	+999.99	-999.99
2's Complement HEX	7FFF	8000

4.8.7 RTD Over Range/Under Range Reading with Modbus RTU protocol

Over Range	Under Range
7FFFh	8000h

4.8.8 Protocol Switching

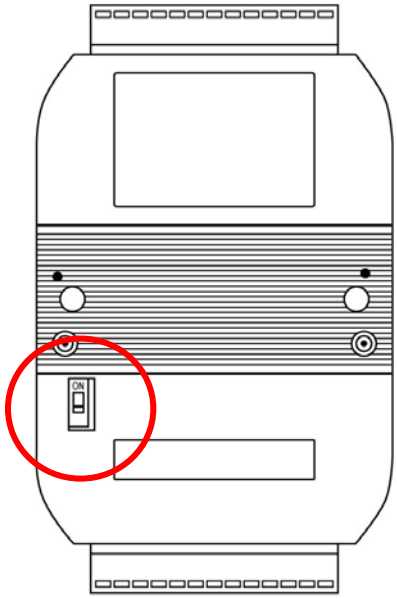
- **To switch to the ASCII protocol**:
  1. Uses sub-function 06h of the function 46h and set byte 8 to a value of 1.
  2. After a power-on reset, the communication protocol will be changed to ASCII protocol.
- **To switch to the Modbus RTU protocol**:
  1. Sends the $AAPN command and set N to a value of 1.
  2. After a power-on reset, the communication protocol will be changed to the Modbus RTU protocol.

4.8.9 INIT Mode

When the module is powered on, with the rear slide switch set to the INIT position (**ON** position) as shown in the figure below, the module is in INIT mode, and the communication settings are as follows:

1. Address: 00
2. Baud Rate: 9600 bps
3. No checksum
4. Protocol: ASCII protocol

If communication with the module is not possible, set the module to the INIT mode and use the above settings to communicate with the module. To read the current settings, send the commands $AA2, and $AAP,. To set new settings, send the commands %AANNTTCCFF, and $AAPN. The new communication settings will be effective after the next power-on reset.



4.9 Command description

4.9.1 %AANNTTCCFF Set Module Configuration

Description: set the configuration of the module at address AA.

Syntax: %AANNTTCCFF (cr)

% is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
NN new module address (00~FF)
TT Always be 00
CC represents the baud rate code.
FF is a hexadecimal number that equals the 8-bit parameter that represents the data format, checksum status and integration time. Bits 2 to 5 are not used, and are set to 0.
(cr) is the terminating character, carriage return (0Dh).
(Please refer to Sec-4.8 to see TT, CC and FF parameter definition)

Response: !AA (cr) if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax error or communication error.
! delimiter character indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
(cr) is the terminating character, carriage return (0Dh).

Example: Command: **%02030006 02(cr)**
Response: !02(cr)
Result: ew module address=03
baud rate=06 (9600)
data format =02 (2's complement hexadecimal)

4.9.2 #** Synchronized sampling

Description: hen the command is received; it will allow all analog input modules to read data from all input channels and will store the data for later retrieval.

Syntax: #**(cr)
Delimiter character
** Synchronized sampling command

Response: There is no response with this command. To access the data, another command, $AA4, must be sent

Example: command: #**
Response: No response
Send the synchronized sampling command.

Example: command: $014
Response: >011+025.123+027.923+045.153+015.020-025.123+015.143
Send a command to read the synchronized data. The status byte of the response is 1, which means that it is the first time the synchronized data has been read after the previous #** command.

Example: command: $014
Response: >010+025.123+027.923+045.153+015.020-025.123+015.143
Send a command to read the synchronized data. The status byte of the response is 0, which means that it is **NOT** the first time the synchronized data has been read after the previous #** command.

4.9.3 #AA Read analog data

Description: The command will return the input value from a specified (AA) module in the currently configured data format.

Syntax: AA(cr)

is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
(cr) is the terminating character, carriage return (0Dh).

Response: (data)(cr) if the command is valid or ?AA (cr) if the command is invalid
There is no response if the module detects a syntax error or communication error.
> is a delimiter character.
(data) is the input value in the configured data format of the module.
(cr) is the terminating character, carriage return (0Dh).

Example: Command: #21(cr)
Response: >+7.2111+7.2567+7.3125+7.1000+7.4712+7.2555+7.1234+7.5678(cr)
The command response the analog input module at address 21h for its input values of all channels. The analog input module responds with channels from 0 to 7 with +7.2111 volts, +7.2567 volts, +7.3125 volts, +7.1000 volts, +7.4712 volts, +7.2555 volts, +7.1234 volts and +7.5678 volts.

Example: Command: #DE(cr)

Response: >FF5DE4323212AE3323345663E000FF03(cr)

The analog input module at address DEh has an input value of FF5DE4323212AE3323345663E000FF03. (The configured data format of the analog input module is two's complement)

4.9.4 #AAN Read analog input from channel N

Description: The command will return the input value from one of the eight channels of a specified (AA) module in the currently configured data format.

Syntax: AAN(cr)

is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the analog input module.
N identifies the channel you want to read. The value can range from 0 to 7
(cr) is the terminating character, carriage return (0Dh).

Response: >(data)(cr) if the command is valid or **?**AA (cr) if the command is invalid
There is no response if the module detects a syntax error or communication error.
> is a delimiter character.
(data) is the input value of the channel number N. Data consists of a + or - sign followed by five decimal digits with a fixed decimal point.
(cr) is the terminating character, carriage return (0Dh).

Example: Command: #120(cr)
Response: >+1.4567(cr)
The command requests the analog input module at address 12h to return the input value of channel 0. The analog input module responds that the input value of channel 0 is equal to +1.4567 volts.

4.9.5 $AA0 Span calibration

Description: Calibrates an analog input module to correct for gain errors.

Syntax: $AA0(cr)

$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the module which is to be calibrated.
0 **is the Span Calibration command.**
(cr) is the terminating character, carriage return (0Dh).

Response: (cr) if the command is valid or ?AA (cr) if the command is invalid
There is no response if the module detects a syntax error or communication error.
! delimiter character indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of the module.
(cr) represents terminating character, carriage return (0Dh).
In order to successfully calibrate an analog input module's input range, a proper calibration input signal should be connected to the analog input module before and during the calibration.

4.9.6 $AA1 Offset calibration

Description: Calibrates an analog input module to correct for offset errors.
Syntax: $AA1(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of the module you want to calibrate.
1 is the Offset Calibration command.
(cr) is the terminating character, carriage return (0Dh).
Response: !AA(cr) if the command is valid or ?AA (cr) if the command is invalid.
here is no response if the module detects a syntax error or communication error.
! delimiter character indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of the module.
(cr) represents terminating character, carriage return (0Dh).

4.9.7 $AA2 Read configuration status

Description: Requests the return of the configuration data from the module at address AA.
Syntax: $AA2(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
2 is the Configuration Status command.
(cr) is the terminating character, carriage return (0Dh).
Response: !AATTCCFF(cr) if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax error or communication error.
! delimiter character indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of an analog input module.
TT the type of channel 0 (Please refer to $AA8Ci command in Sec.4.9.13).
CC represents the baud rate code.
FF is a hexadecimal number that equals the 8-bit parameter that represents the data format, checksum status and integration time.Bits 2 to 5 are not used, and are set to 0.
(cr) is the terminating character, carriage return (0Dh).
(Please refer to Sec-4.8 to see TT, CC and FF parameter definition)
Example: Command: $452(cr)
Response: !45050600(cr)
The command asks the analog input module at address 45h to send its configuration data. The analog input module at address 45h responds with an input range of 2.5 volts, a baud rate of 9600 bps, an integration time of 50 ms (60 Hz), engineering units are the currently configured data format, and no checksum function or checksum generation.

4.9.8 $AA4 Read Synchronized data

Description: Reads the synchronized data that was retrieved by the last #** command.

Syntax: $AA4(cr)

$ Delimiter character
AA Address of the module to be read (00 to FF)
4 Command to read the synchronized data

Response: >AAS(Data)(cr) or Invalid Command: ?AA(cr)

! Delimiter character for a valid command
? Delimiter character for an invalid command
AA Address of the responding module (00 to FF)
S Status of the synchronized data
1: first read
0: not the first read
(Data) Synchronized data. See Section data format. The data from disabled channels is filled with 7 space characters.

There will be no response if the command syntax is incorrect, there is a communication error.

Examples: Command: $014
Response: ?01
Reads the synchronized data for module 01. An invalid command is returned if the #**command has not been issued in advance.
Command: #**
Response: No response
Sends the synchronized sampling command.
Command: $014
Response: >011+025.123+027.923+045.153+015.020-025.123+015.143
Reads the synchronized data for module 01. The module returns the synchronized data and sets the status byte to 1 to indicate that the synchronized data is the first time be read.
Command: $014
Response: >010+025.123+027.923+045.153+015.020-025.123+015.143
Reads the synchronized data for module 01. The module returns the synchronized data and sets the status byte to 0 to indicate that the synchronized data has been read.

4.9.9 $AA5 Read the module reset status

Description: Reads the reset status of a module.

Syntax: $AA5(cr)

$ Delimiter character
AA Address of the module to be read (00 to FF)
5 Command to read the module reset status

Response: !AAS(cr) or Invalid Command: ?AA(cr)

! Delimiter character for a valid command
? Delimiter character for an invalid command
AA Address of the responding module (00 to FF)
S Reset status of the module
1: This is the first time the command has been sent since the module was powered on.
0: This is not the first time the command has been sent since the module was powered on, which denotes that there has been no module reset since the last $AA5 command was sent.

There will be no response if the command syntax is incorrect, there is a communication error.

Examples: Command: $015
Response: !011
Reads the reset status of module 01. The response shows that it is the first time the $AA5 command has been sent since the module was powered-on.
Command: $015
Response: !010
Reads the reset status of module 01. The response shows that there has been no module reset since the last $AA5 command was sent.

4.9.10 $AA5VV Enable/disable channels for multiplexing

Description: Enables/disables multiplexing simultaneously for separate channels of a specified input module.

Syntax: $AA5VV(cr)

- $ is a delimiter character.
- AA (range 00-FF) represents the 2-character hexadecimal address of module.
- **5** is the Enable/disable Channels command.
- VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4~7, and the second word represents the status of channel 0~3. Value 0 means the channel is disabled, value 1 means the channel is enabled.
- (cr) is the terminating character, carriage return (0Dh).

Response: !AA(cr) if the command is valid or ?AA (cr) if the command is invalid.

There is no response if the module detects a syntax error or communication error.

- ! delimiter character indicates a valid command was received.
- ? delimiter character indicates the command was invalid.
- AA (range 00-FF) represents the 2-character hexadecimal address of module.
- (cr) is the terminating character, carriage return (0Dh).

Example: Command: $00581(cr)

Response:!00(cr)

Hexadecimal 8 equals binary 1000, which enables channel 7 and disables channels 4, 5, and 6.

Hexadecimal 1 equals binary 0001, which enables channel 0 and disables channel 1, 2, and 3.

4.9.11 $AA6 Read channel status

Description: Asks a specified input module to return the status of all channels.

Syntax: $AA6(cr)

- AA (range 00-FF) represents the 2-character hexadecimal address of the module of which the channel status you want to send. The channel status defines whether a channel is enabled or disabled.
- 6 is the Read Channel Status command.
- (cr) is the terminating character, carriage return (0Dh).

Response: !AAVV(cr) if the command is valid or ?AA (cr) if the command is invalid.

There is no response if the module detects a syntax error or communication error.

- ! delimiter character indicates a valid command was received.
- ? delimiter character indicates the command was invalid.
- AA (range 00-FF) represents the 2-character hexadecimal address of module.
- VV are two hexadecimal values. The values are interpreted by the module as two binary words (4-bit). The first word represents the status of channel 4-7, and the second word represents the status of channel 0-3. Value 0 means the channel is disabled, value 1 means the channel is enabled.
- (cr) is the terminating character, carriage return (0Dh).

Example: Command: $026(cr)

Response: !02FF(cr)

The command asks the analog input module at address 02 to send the status of it input channels. The analog input module at address 02 responds that all its multiplex channels are enabled (FF equals 1111 and 1111).

4.9.12 $AA7CiRrr Set channel type individually

Description: The command set channel type individually.
Syntax: $AA7CiRrr(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
7C is the Set channel type command.
i channel number
rr channel type code
(cr) is the terminating character, carriage return (ODh).
Response: !AA if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax error or communication error.
! is a delimiter character indicating a valid command was received.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
(cr) is the terminating character, carriage return
Examples Command: $017C3R08(cr)
Response: !01(cr)
Set type code 08 (+/-10V) to channel 3.

4.9.13 $AA8Ci Read individual channel type

Description The command read individual channel type.
Syntax $AA8Ci (cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
8C is the read channel type command.
i channel number
(cr) is the terminating character, carriage return (ODh).
Response: !AACiRrr if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax error or communication error.
! is a delimiter character indicating a valid command was received.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
i channel number(0~7)
rr type of channel **i**
(cr) is the terminating character, carriage return
Examples: Command: $018C3(cr)
Response: !01C3R08(cr)
The type code of channel 3 is 08 (+/-10V).

4.9.14 $AAB Read Channel Burnout Status

Description: Read channel burn out status
Syntax: $AAB(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
B is the Channel Diagnose command.
(cr) is the terminating character, carriage return (0Dh).
Response: !AANN(cr) if the command is valid when it applied with.
?AA(cr) if an invalid command was issued.
There is no response if the module detects a syntax error or communication error.
! delimiter character indicates a valid command was received.
? delimiter character indicates the command was invalid.
AA (range 00-FF) represents the 2-character hexadecimal address of the module.
NN (range 00-FF) is a hexadecimal number that equals the 8-bit parameter, representing the status of analog input channels. Bit value 0 means normal status; and bit value 1 means channel open wiring.
(cr) is the terminating character, carriage return (0Dh)
Examples: Command: $01B(cr)
Response: !0101(cr)
Channel 0 is open wiring and channel 1~7 are all normal.

4.9.15 $AAF Read the firmware version

Description: The command requests the module at address AA to return the version code of its firmware.

Syntax: $AAF(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
F is the Read Firmware Version command.
(cr) is the terminating character, carriage return (ODh).

Response: !AA(Version)(cr) if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax error or communication error.
! is a delimiter character indicating a valid command was received.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
(Version) is the version code of the module's firmware at address AA.
(cr) is the terminating character, carriage return

4.9.16 $AA1 Read the INIT terminal status

Description: Performs a zero calibration.

Syntax: $AA1(cr)
$ Delimiter character
AA Address of the module to be calibrated (00 to FF)
1 Command for the zero calibration

Response: !AA(Version)(cr) if the command is valid or ?AA (cr) if the command is invalid.
! Delimiter character for a valid command
? Delimiter character for an invalid command
AA Address of the responding module (00 to FF)
There is no response if the module detects a syntax error or communication error.

Examples: Command: $011
Response: !01
Performs a zero calibration on module 01 and returns a valid response.

Examples: Command: $021
Response: ?02
Performs a zero calibration on module 02. An invalid command is returned because the "enable calibration" command was not sent in advance. (see ~AAEV command)

4.9.17 $AAM Read the module name

Description: Requests the analog output module at address AA to return its name

Syntax: $AAM(cr)
$ is a delimiter character.
AA (range 00-FF) represents the 2-character hexadecimal address that you want to access.
M is the Read Module Name command.
(cr) is the terminating character, carriage return (ODh)

Response: !AA(Module Name)(cr) if the command is valid or ?AA (cr) if the command is invalid.
There is no response if the module detects a syntax or communication error.
! is a delimiter character indicating that a valid command was received.
AA (range 00-FF) represents the 2-character hexadecimal address of module.
(Module Name) is the name of the module at address AA.
(cr) is the terminating character, carriage return

4.9.18 $AAP Read the communication protocol

Description: Reads the communication protocol information.
Syntax: $AAP(cr)
$ Delimiter character
AA Address of the module to be read (00 to FF)
P Command to read the communication protocol
Response: !AASC (cr) or Invalid Response: ?AA(cr)
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA Address of the responding module (00 to FF)
S The protocols supported by the module
0: onlyASCII protocol is supported
1: both the ASCII and Modbus RTU protocols are supported
C Current protocol saved in EEPROM that will be used at the next power on reset
0: the protocol set in EEPROM is ASCII protocol
1: the protocol set in EEPROM is Modbus RTU
There is no response if the module detects a syntax or communication error.
Examples: Command: $01P
Response: !0110
Reads the communication protocol of module 01 and
returns a response of 10 meaning that it supports both the ASCII and Modbus RTU protocols and the protocol that will be used at the next power on reset is ASCII protocol.

4.9.19 $AAPN Set the communication protocol

Description: Sets the communication protocol.
Syntax: $AAPN(cr)
$ Delimiter character
AA Address of the module to be read (00 to FF)
P Command to set the communication protocol
N 0: ASCII protocol,1: Modbus RTU protocol
The new protocol is saved in the EEPROM and will be effective after the next power on reset.
Response: !AA (cr) or Invalid Response: ?AA (cr)
! Delimiter character for a valid response
? Delimiter character for an invalid response
AA Address of the responding module (00 to FF)
There is no response if the module detects a syntax or communication error.
Examples: Command: $01P1
Response: !01
Sets the communication protocol of module 01 to Modbus RTU and returns a valid response.

4.9.20 $AAS1 Reload default calibration parameters

Description: Reloads the factory default calibration parameters, including the internal calibration parameters.
Syntax: $AAS1(cr)
$ Delimiter character
$ Delimiter character
AA Address of the module to have the default parameters reloaded (00 to FF)
S1 Command to reload the factory default calibration parameters
Response: Valid command: !AA(cr) or Invalid command: ?AA(cr)
! Delimiter character for a valid command
? Delimiter character for an invalid command
AA Address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.
Examples: Command: $01S1
Response: !01
Sends the command to reload the factory default calibration parameters for module 01 and returns a valid response.

4.9.21 ~AAD Read the miscellaneous settings

Description: Reads the miscellaneous settings of a module.

Syntax: ~AAD(cr)

~ Delimiter character

AA Address of the module to be read (00 to FF)

D Command to read the miscellaneous settings

Response: !AAVV(cr) or Invalid Command: ?AA(cr)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA Address of the responding module (00 to FF)

VV Two hexadecimal digits that represent the miscellaneous settings as follows:

7	6	5	4	3	2	1	0
Reserved				SU	Reserved		

SU(bit 3): Setting for under range reading

Bit 3=0 the reading of under range is as usual

Bit 3=1 force the reading of under range to be the same as over range

Examples: Command: ~01D

Response: !0108

Reads the miscellaneous settings of module 01 and returns 08.

4.9.22 ~AADVV Set the miscellaneous settings

Description: Set the miscellaneous settings of a module.

Syntax: ~AADVV(cr)

~ Delimiter character

AA Address of the module to be read (00 to FF)

D Command to read the miscellaneous settings

VV Two hexadecimal digits that represent the miscellaneous settings as follows:

7	6	5	4	3	2	1	0
Reserved				SU	Reserved		

SU(bit 3): Setting for under range reading

Bit 3=0 the reading of under range is as usual

Bit 3=1 force the reading of under range to be the same as over range

Response: !AA(cr) or Invalid Command: ?AA(cr)

! Delimiter character for a valid command

? Delimiter character for an invalid command

AA Address of the responding module (00 to FF)

Examples: Command: ~01D08

Response: !01

Sets the miscellaneous settings of module 01 to 08, and returns a valid response.

4.9.23 ~AAEV Enable/disable calibration

Description: Enable/Disable module calibration.
Syntax: ~AAEV(cr)
~ delimiter character
AA address of the module
E command to enable/disable calibration
V 1: enable calibration, 0: disable calibration
Response: !AA(cr) if the command is valid or ?AA (cr) if the command is invalid
! delimiter character for a valid response
? delimiter character for an invalid response
AA address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error
Examples: Command: $010(cr)
Response: ?01(cr)
Send the command to perform a span calibration on module 01. It returns an invalid response because the "enable calibration command" was not sent in advance
Command: ~01E1(cr)
Response: !01(cr)
Enable calibration on module 1
Command: $010(cr)
Response: !01(cr)
Send the command to perform a span calibration on module 01 and returns a valid response.

4.9.24 ~AAI Soft INIT

Description: The Soft INIT command is used to enable modification of the Baud Rate and checksum settings using software only.
Description: Requests the analog output module at address AA to return its name
~AAI(cr)
~ Delimiter character
AA Address of the module to be set (00 to FF)
I Command to set the Soft INIT
Response: !AA(cr) or Invalid command: ?AA(cr)
! Delimiter character for a valid command
? Delimiter character for an invalid command
AA Address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error
Examples: Command: ~01I
Response: !01
Sets the soft INIT of module 01 and returns a valid response.

4.9.25 ~AAO(Name) Set the module name

Description: Set module name
Syntax: ~AAO(name)(cr)
~ delimiter character
AA address of the module to be set (00 to FF)
O **c**ommand to set the module name (new name of the module (max. 6 characters).
Response: !AA(cr) if the command is valid or ?AA (cr) if the command is invalid
! delimiter character for a valid response
? delimiter character for an invalid response
AA address of the responding module (00 to FF)
There will be no response if the command syntax is incorrect, there is a communication error

4.9.26 ~AATnn Set the soft INIT timeout

Description: Sets the soft INIT timeout value.
Syntax: ~AATnn(cr)
~ Delimiter character
AA Address of the module to be set (00 to FF)
T Command to set the soft INIT timeout
nn Two hexadecimal digits representing the timeout value in seconds. The maximum timeout value is 60 seconds. When changing the Baud Rate or checksum settings without altering the INIT* pin, the ~AAI and %AANNTTCCFF commands should be sent consecutively and the time interval between the two commands should be less than the soft INIT timeout. If the soft INIT timeout is 0, then the Baud Rate and checksum settings cannot be changed using software only. The power-on reset value of the soft INIT timeout is 0.
There will be no response if the command syntax is incorrect, there is a communication error

4.9.27 ~** Sends a "Host OK"

Description: Informs all modules that the host is OK.
Syntax: ~**(cr)
~ delimiter character
** Host OK command
Response: no response

4.9.28 ~AA0 Read the host watchdog status

Description: Reads the host watchdog status of a module.
Syntax: ~AA0(cr)
~ delimiter character
AA address of the module to be read (00 to FF)
0 command to reads the host watchdog status.
Response: !AASS(cr) if the command is valid or ?AA(cr) if the command is invalid
! delimiter character for a valid response
? delimiter character for an invalid response
AA address of the responding module (00 to FF)
SS two hexadecimal digits that represent the host watchdog status,
where:
Bit 7: =0 indicates that the host watchdog is disabled and
=1 indicates the host watchdog is enabled,
Bit 2: =1 indicates watchdog time out has occurred.
The host watchdog status is stored in EEPROM and can only be reset using the ~AA1 command.

4.9.29 ~AA1 Reset the host watchdog status

Description: Reset the host watchdog time out status of a module.
Syntax: ~AA1(cr)
~ delimiter character A
AA address of the module to be set (00 to FF)
1 command to reset the host watchdog time out status
Response: !AA (cr) if the command is valid or ?AA (cr) if the command is invalid
! delimiter character for a valid response module w
? delimiter character for an invalid response le (00 to FF)
AA address of the responding module
There will be no response if the command syntax is incorrect, there is a communication error

4.9.30 ~AA2 Read the host watchdog time out value

Description: Reads the host watchdog time out value

Syntax: ~AA2(cr)

~ delimiter character
AA address of the module to be read (00 to FF)
2 command to read the host watchdog time out value.

Response: !AAEVV(cr) if the command is valid or ?AA (cr) if the command is invalid

! delimiter character for a valid response
? delimiter character for an invalid response
AA address of the responding module
E 1: the host watchdog is enabled
0: the host watchdog is disabled V
VV two hexadecimal digits to represent the value in tenths of a second,
There will be no response if the command syntax is incorrect, there is a communication error

Examples: Command: ~012(cr)
Response: !011FF(cr)
Reads the host watchdog time out value of module 01 and returns FF, meaning that the host watchdog is enabled and watchdog timeout value is FF(hex)=255.5 seconds

4.9.31 ~AA3ETT Enables/disables the host watchdog

Description: Enables/disables the host watchdog and set the host watchdog time out value of a module.

Syntax: ~AA3EVV(cr)

~ delimiter character
AA address of the module to be read (00 to FF)
3 command to set watchdog time out value
E 1: enable host watchdog
0: disable host watchdog
TT two hexadecimal digits to represent timeout value in tenths of second, for example, 01 means 0.1 seconds and FF means 25.5 seconds

Response: !AA(cr) if the command is valid or ?AA (cr) if the command is invalid

! delimiter character for a valid response
? delimiter character for an invalid response
AA address of the responding module
There will be no response if the command syntax is incorrect, there is a communication error

Examples: Command**:** ~013164(cr)
Response: !01(cr)
Enable host watchdog of module 01 and sets e host watchdog time out value to 10.0 seconds.

Chapter 5 Modbus RTU Command structure

EDAM-8015 system accepts a command/response form with the host computer. When systems are not transmitting they are in listen mode. The host issues a command to a system with a specified address and waits a certain amount of time for the system to respond. If no response arrives, a time-out aborts the sequence and returns control to the host. This chapter explains the structure of the commands with Modbus RTU protocol, and guides to use these command sets to implement user's programs.

5.1 Modbus Function code introductions

Code (Hex)	Name	Usage
46	Module Misc. configuration	Module Misc. configuration
01	Read discrete coil	Read channel burnout status
02	Read discrete input	Read channel burnout status
03	Read Holding Registers	Read 16-bit register. Used to read integer or floating point process data.
04	Read Input Registers	
05	Write single coil	
06	Preset Single Register	Write data in 16-bit integer format
0F	Write multiple coils	
10	Preset Multiple Registers	Write multiple data in 16-bit integer format

5.2 Modbus Function code 0x46 for Module configuration

This function code 0x46 is used to read or change the settings of the module. The following sub-function codes are supported.

Sub-function Code	Description	Sec.
00 (0x00)	Read the module name	5.2.1
04 (0x04)	Set the module address	5.2.2
05 (0x05)	Read the communication settings	5.2.3
06 (0x06)	Set the communication settings	5.2.4
07 (0x07)	Read the type code	5.2.5
08 (0x08)	Set the type code	5.2.6
32 (0x20)	Read the firmware version	5.2.7
37 (0x25)	Read the channel enable/disable status	5.2.8
38 (0x26)	Set the channel enable/disable	5.2.9
41 (0x29)	Read the miscellaneous settings	5.2.10
42 (0x2A)	Write the miscellaneous settings	5.2.11

5.2.1 Sub-function 00 (0x00) Read module name

Description: read the name of a module.

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x00

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x00
03 ~ 06	Module name	4 Bytes	0x00 0x80 0x15 0x00

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes received

5.2.2 Sub-function 04 (0x04) Set module address

Description: set module address

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x04
03	New address	1 Byte	1 to 255
04 ~ 06	Reserved	3 Bytes	0x00 0x00 0x00

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x04
03	Set address result	1 Byte	0: OK, others: error
04 ~ 06	Reserved	3 Bytes	0x00 0x00 0x00

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: new address out of range, reserved bytes are not filled with zero, or incorrect number of bytes

5.2.3 Sub-function 05 (0x05) Read communication settings

Description: read communication protocol

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x05
03	Reserved	1 Byte	0x00

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x05
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0x03 ~ 0x0A, Baud Rate code, see 4.8.2 for details.
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0x00: no parity, 1 stop bit
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: ASCII protocol 1: Modubs RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: reserved bytes are not filled with zero, or incorrect number of bytes

5.2.4 Sub-function 06 (0x06) Set communication settings

Description: set communication protocol

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0x03 ~ 0x0A, Baud Rate code, see 4.8.2 for details.
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0x00: no parity, 1 stop bit
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: ASCII protocol 1: Modubs RTU protocol
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0: OK, others: error
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0: OK, others: error
07	Reserved	1 Byte	0x00
08	Mode	1 Byte	0: OK, others: error
09 ~ 10	Reserved	2 Bytes	0x00 0x00

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: reserved bytes are not filled with zero, or incorrect number of bytes

Note:

This new settings are all saved into EEPROM and will be used for the next power-on

5.2.5 Sub-function 07(0x07) Read channel type code

Description: read analog channel type code

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x07
03	Reserved	1 Bytes	0x00
04	Channel	1 Byte	0x00 ~ 0x05

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x07
03	Type code	1 Byte	Type code, see Se.4.8.3 for details

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: reserved bytes are not filled with zero, or incorrect number of bytes

5.2.6 Sub-function 08(0x08) Set channel type code

Description: set analog channel type code

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x08
03	Reserved	1 Bytes	0x00
04	Channel	1 Byte	0x00 ~ 0x05
05	Type code	1 Byte	Type code, see Se.4.8.3 for details

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x07
03	Type code	1 Byte	0: OK, others: error

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: type code is out of range, reserved bytes are not filled with zero, or incorrect number of bytes

5.2.7 Sub-function 32(0x20) Read firmware version

Description: read firmware version

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x20

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x20
03	Major version	1 Byte	0x00 ~ 0xFF
04	Minor version	1 Byte	0x00 ~ 0xFF
05	Build version	1 Byte	0x00 ~ 0xFF

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes

5.2.8 Sub-function 37(0x25) Read channel enable/disable status

Description: read channel enable/disable status

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x25

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x25
03	Enabled/disabled status	1 Byte	0x00 ~ 0x3F, enabled/disabled status of each channel, bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it denotes that the channel is enabled and 0 denotes that the channel is disabled

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes

5.2.9 Sub-function 38(0x26) Set channel enable/disable

Description: set channel enable/disable

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x26
03	Enabled/disabled status	1 Byte	0x00 ~ 0x3F, enabled/disabled settings of each channel, bit 0 corresponds to channel 0, bit 1 corresponds to channel 1, etc. When the bit is 1 it denotes that the channel is enabled and 0 denotes that the channel is disabled

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x26
03	Enabled/disabled status	1 Byte	0: OK, others: error

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes

5.2.10 Sub-function 41(0x29) Read miscellaneous settings

Description: read miscellaneous settings

Request:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x29

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x29
03	Miscellaneous settings	1 Byte	Bit 7: filter setting 0: 60Hz rejection 1: 50Hz rejection Bit 6~0: reserved

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes

5.2.11 Sub-function 42(0x2A) Set miscellaneous settings

Description: set miscellaneous settings

Request:

L

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2A
03	Miscellaneous settings	1 Byte	Bit 7: filter setting 0: 60Hz rejection 1: 50Hz rejection Bit 6~0: reserved

Response:

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0x46
02	Sub function code	1 Byte	0x2A
03	Miscellaneous settings	1 Byte	0: OK, others: error

Error Response

00	Address	1 Byte	1 to 255
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	03: incorrect number of bytes

5.3 Modbus Address Mapping

5.3.1 Discrete address (1-bits)

Discrete address mapping for Modbus function code 0x01, 0x02, 0x05, 0x0F

Address		Channel	Description	Attribute
Hex	Decimal			
0080 ~ 0085	00128 ~ 00133	0~5	Read over/under range status A bit corresponds to a channel. The bit=1 it denotes that the channel is over/under range. The bit=0 it denotes that the channel is disabled or normal	R
0100	00256		Read/write protocol 0: ASCII , 1: Modbus protocol	R/W
0102	00258		Read/Write Filter setting, 0: 60Hz, 1: 50Hz	R/W
0104	00260		1: Enable, 0: disable host watchdog	R/W
010C	00268		Modbus data format, 0: Hex, 1:enginerring	
010D	00269		Read host watchdog timeout status Write 1 to clear host watchdog timeout status	R/W
010F	00271		Write 1 to load factory default calibration parameters	R/W
0110	00272		Rest status: 1: first read after powered-on 0: not the first read after powered-on	R
0112	00274		1: force to return 7FFF for wire opening	R/W

5.3.2 Register address (16-bits)

Input/Holding register address mapping for Modbus function code 0x03, 0x04, 0x06, 0x10

Address		Channel	Description	Attribute
Hex	Decimal			
0000 ~ 0005	00000 ~ 00005	0~5	Read analog input value	R
0100 ~ 0105	256 ~ 261	0~5	Read/set type of channel 0 to 5	R/W
0120 ~ 0125	288 ~ 293	0~5	Read/set temperature offset of channel 0 to 5 in 0.1℃ valid range from -128(FF80) to 127(007F)	R/W
180 ~ 185	384 ~ 389	0~5	RTD resistance offset of channel 0 to 5 in 0.1 ohms, range from 0~255	R/W
1E0	480		Firmware version (low word)	
1E1	481		Firmware version (high word)	R/W
1E2	482		Module name(low word)	R/W
1E3	483		Module name(high word)	R
1E4	484		Read/write Module address (range from 1~255)	R/W
1E5	485		Read/write baud rate bit 5:0	R/W
1E7	487		Read/write Modbus response delay time in ms, range from 0 to 30	R/W
1E8	488		Read/write host watchdog timeout value, 0~255 in 0.1 second	R/W
1E9	489	0~5	Read/Write Channel enable/disable status	R/W
1EB	491		Read host watchdog timeout counts Write 0 to clear counter	R/W

5.4 Modbus Engineering Data Format Table

The Modbus engineering format is that temperature value is engineering formatted in 0.1°C
formula: Temperature=Modbus data /10 °C

Type Code	RTD Type	Min.(Hex)	Max. (Hex)
20	Platinum 100 α= 0.00385 (-100 ~ 100°C)	FC18	03E8
21	Platinum 100 α= 0.00385 (0 ~ 100°C)	0000	03E8
22	Platinum 100 α= 0.00385 (0 ~ 200°C)	0000	07D0
23	Platinum 100 α= 0.00385 (0 ~ 600°C)	0000	1770
24	Platinum 100 α = 0.003916 (-100 ~ 100°C)	FC18	03E8
25	Platinum 100 α = 0.003916 (0 ~ 100°C)	0000	03E8
26	Platinum 100 α = 0.003916 (0 ~ 200°C)	0000	07D0
27	Platinum 100 α = 0.003916 (0 ~ 600°C)	0000	1770
28	Nickel 120 (-80 ~ 100°C)	FCE0	03E8
29	Nickel 120 (0 ~ 100°C)	0000	03E8
2A	Platinum 1000 α= 0.00385 (-200 ~ 600°C)	F830	1770
2B	Cu 100 α= 0.00421 (-20 ~ 150°C)	FF38	05DC
2C	Cu 100 α= 0.00427 (0 ~ 200°C)	0000	2000
2D	Cu 1000 α = 0.00421 (-20 ~ 150°C)	FF38	05DC
2E	Platinum 100 α = 0.00385 (-200 ~ 200°C)	F830	07D0
2F	Platinum 100 α= 0.003916 (-200 ~ 200°C)	F830	07D0
80	Platinum 100 α= 0.00385 (-200 ~ 600°C)	F830	1770
81	Platinum 100 α= 0.003916 (-200 ~ 600°C)	F830	1770
82	Cu 50 (-50 ~ 150°C)	FE0C	05DC
83	Nickel 100 (-60 ~ 180°C)	FDA8	0708

Example: Assume the Modbus data=0x0240(Hex)=576(Dec)
The temperature of the specified channel is 576/10=57.6°C
Example: Assume the Modbus data=0xEF1B(Hex)= - 4325(Dec)
The voltage of channel 2 is -4235/10=423.5°C

5.5 Modbus Hex 2's complement Data Format Table

Type Code	RTD Type	Min.	Max.	Formula
20	Platinum 100 α= 0.00385 (-100 ~ 100°C)	8001	7FFF	Temp.=(Modbus data*100) /32767 °C
21	Platinum 100 α= 0.00385 (0 ~ 100°C)	0000	7FFF	Temp.=(Modbus data*100) /32767 °C
22	Platinum 100 α= 0.00385 (0 ~ 200°C)	0000	7FFF	Temp.=(Modbus data*200) /32767 °C
23	Platinum 100 α= 0.00385 (0 ~ 600°C)	0000	7FFF	Temp.=(Modbus data*600) /32767 °C
24	Platinum 100 α= 0.003916 (-100 ~ 100°C)	8001	7FFF	Temp.=(Modbus data*100) /32767 °C
25	Platinum 100 α= 0.003916 (0 ~ 100°C)	0000	7FFF	Temp.=(Modbus data*100) /32767 °C
26	Platinum 100 α= 0.003916 (0 ~ 200°C)	0000	7FFF	Temp.=(Modbus data*200) /32767 °C
27	Platinum 100 α= 0.003916 (0 ~ 600°C)	0000	7FFF	Temp.=(Modbus data*600) /32767 °C
28	Nickel 120 (-80 ~ 100°C)	999B	7FFF	Temp.=(Modbus data*100) /32767 °C
29	Nickel 120 (0 ~ 100°C)	0000	7FFF	Temp.=(Modbus data*100) /32767 °C
2A	Platinum 1000 α= 0.00385 (-200 ~ 600°C)	D556	7FFF	Temp.=(Modbus data*600) /32767 °C
2B	Cu 100 α= 0.00421 (-20 ~ 150°C)	EEF0	7FFF	Temp.=(Modbus data*150) /32767 °C
2C	Cu 100 α= 0.00427 (0 ~ 200°C)	0000	7FFF	Temp.=(Modbus data*200) /32767 °C
2D	Cu 1000 α= 0.00421 (-20 ~ 150°C)	EEF0	7FFF	Temp.=(Modbus data*150) /32767 °C
2E	Platinum 100 α= 0.00385 (-200 ~ 200°C)	8001	7FFF	Temp.=(Modbus data*200) /32767 °C
2F	Platinum 100 α= 0.003916 (-200 ~ 200°C)	8001	7FFF	Temp.=(Modbus data*200) /32767 °C
80	Platinum 100 α= 0.00385 (-200 ~ 600°C)	D556	7FFF	Temp.=(Modbus data*600) /32767 °C
81	Platinum 100 α= 0.003916 (-200 ~ 600°C)	D556	7FFF	Temp.=(Modbus data*600) /32767 °C
82	Cu 50 (-50 ~ 150°C)	D556	7FFF	Temp.=(Modbus data*150) /32767 °C
83	Nickel 100 (-60 ~ 180°C)	D556	7FFF	Temp.=(Modbus data*180) /32767 °C

Example: Assume type of channel 2 is 2E and Modbus data=0x2030(Hex)=8240(Dec)
The temperature of channel 2 is (8240*200)/32767=50.294 °C

Example: Assume type of channel 1 is 2E and Modbus data=0xC001(Hex)=-16383(Dec)
The voltage of channel 2 is (-16383*200)/32767=-99.996 °C